

Lucee on Jetty v0.7 Documentation

Lucee on Jetty is a project to provide a Lucee-enabled Jetty instance with sensible configuration and comprehensive documentation.

See the file **readme.md** for project information, requirements, licensing and credits.

This documentation aims to be a comprehensive guide to Lucee on Jetty - please raise an issue if you find any bugs/errors/omissions, or have any questions.

Hyperlinks that are internal to the PDF are [blue and dashed](#), those to external websites are [red and underlined](#).

Lucee on Jetty Documentation

Introduction

[Lucee on Jetty](#) is a project to provide a Lucee package that is easy to use and provides a fully functional & capable web server, with extensive documentation on how everything has been done, so it can be understood and adapted as needed.

This documentation aims to be a comprehensive guide to Lucee on Jetty, please [raise an issue](#) if you find any bugs/errors/omissions, or have any questions.

Understanding Terms

There is a [Glossary](#) which attempts to explain any technical terms used by Lucee and/or Jetty.

Other Documentation

This documentation covers topics relevant to Lucee on Jetty, and will generally provide links to pertinent pages in the respective docs for Jetty or Lucee that may cover a subject in more detail.

The documentation homepages for these are:

- Jetty: <http://www.eclipse.org/jetty/documentation/current/>
- Lucee: <https://bitbucket.org/lucee/lucee/wiki/Home>

A Note on Jetty Documentation

Jetty is very flexible software with multiple ways to achieve the same tasks, and different methods of configuration. This is due to the multitude of use cases Jetty has (including embedded within applications), and the abundance of choice can be reflected in its docs, with some pages containing multiple examples of different methods for doing the same thing, sometimes without it being clear which option is the best.

To confirm how Lucee on Jetty works, or for clarification of what something means or how to achieve something, feel free to [raise an issue](#).

Starting the Server

Running Lucee on Jetty is as easy as unzipping and running a start-server script.

- On Windows, double-click the `start-server.bat` file.
- All other operating systems, execute `start-server.sh` instead.

Pressing Ctrl-C will shutdown the server.

There are sample service scripts for Linux-based OSES provided by Jetty in `jetty-home/bin/` directory.

Defaults

The HTTP server listens on port 8080 and the default context responds to localhost or 127.0.0.1

Browsing to `http://localhost:8080` will run `lucee-base/webapps/ROOT/index.cfm`

The port can be changed in `lucee-base/start.ini` - changing it requires a Jetty restart.

The default context is configured in `lucee-base/webapps/ROOT.xml` - this file also allows virtual host aliases to be configured.

The default `index.cfm` contains links to Lucee Admins, which are available at `http://localhost:8080/lucee/admin/server.cfm` and `http://localhost:8080/lucee/admin/web.cfm`

Lucee non-default configuration

- The `lucee-server-directory` parameter is set to use `lucee-base/lucee-server` instead of the `lucee-base/lib/ext/lucee-server` default.
- The `lucee-web-directory` parameter is set to use `lucee-base/webinf/{web-context-label}` instead of `lucee-base/webapps/{context}/WEB-INF/lucee`

File & Directory Structure

The key files and directories in a Lucee on Jetty bundle resemble the following:

lucee-on-jetty/	
jetty-home/	Unmodified Jetty distribution.
lucee-base/	jetty-base configured for Lucee.
logs/	Jetty log files.
lucee-server/	Lucee server context and patches.
deploy/	Lucee extension (lex) files.
modules/	
lucee/	Lucee module files.
etc/	
lucee-servlets.xml	Lucee servlet configuration.
rewrite-rules.xml	Rewrite handler configuration.
lucee-webapp-common.xml	Common Jetty context configuration.
lucee-global-webapp-common.xml	Applies common Jetty configuration to all contexts.
lib/	
lucee*.jar	Lucee jar file.
lucee.mod	Lucee module definition.
resources/	
jetty-logging.properties	Jetty logging configuration
webapps/	
ROOT/	Default context's webroot.
index.cfm	
ROOT.xml	Default context configuration.
webinf/	Lucee context configuration and log files.
start.ini	Jetty module enabling and configuration.
README.TXT	
docs.pdf	Lucee on Jetty offline documentation
start-server.bat	Startup batch script (Windows).
start-server.sh	Startup shell script.

jetty-home/

This is where Jetty itself is contained - the files in this directory are unmodified. Refer to [Jetty documentation](#) to understand the contents of this directory.

You can upgrade Jetty by renaming this directory and placing a new jetty-home (or jetty-distribution) in its place. (this will probably just work without issue, but as with all upgrades, performing a verified backup before doing it is recommended.)

lucee-base/

Jetty has the concept of jetty-home and jetty-base, allowing a Jetty distribution (jetty-home) to be used without modification by creating one or more jetty-base locations to extend or override the jetty-home settings without needing to change the jetty-home files.

The best place to understand this is the Jetty documentation: [Managing Jetty Base and Jetty Home](#)

Since this concept is still relatively new, some Jetty documentation may still refer to jetty-home (or rather, `${jetty.home}`) in situations where the equivalent jetty-base location would now be the recommended location.

To help reduce confusion, Lucee on Jetty uses the name `lucee-base` for this directory, but if you prefer you can rename it to `jetty-base` - or indeed any name you like. (There is a single reference to `lucee-base` in each of the `start-server` scripts that should be updated to match the renamed directory, but otherwise nothing else is affected by what it is called.)

The `start-server` scripts work by going into the `lucee-base` directory, then calling `../jetty-home/start.jar` from Java to start Jetty with configuration read from the `start.ini` file.

start.ini

This file is how Jetty is instructed which modules and configuration files to load on startup, and can override any default settings.

Jetty docs: [Jetty Start Configuration Files](#)

The minimum modules required to run a Lucee-enabled Jetty server are `lucee`, `deploy` and `http` - you may add/remove whatever other modules are useful to you.

logs/

This directory is where Jetty will place log files, including `stderr` and `stdout`, when the `console-capture` module is enabled.

For details on Jetty logging, see the Jetty documentation: <http://www.eclipse.org/jetty/documentation/current/configuring-logging.html>

Lucee log files are located within a web context's configuration directory, that is, within `lucee-base/webinf/{web-context-label}/logs`

lucee-server/

This `lucee-server` directory is where Lucee stores the configuration for the Lucee Server and the patches when Lucee is updated - by default this directory is found in `lib/ext/lucee-server` (alongside the Lucee JARs), but this bundle has been configured for a more convenient location.

The context directory contains the `lucee-server.xml` config file for the Lucee Server Admin, along with other global/server-level files.

The patches directory will contain one or more `X.X.X.xxx.lco` files - the latest patch in this directory is the version Lucee will run.

A manual upgrade can be performed by copying a newer `.lco` file to this directory and restarting Lucee.

lucee-server/deploy/

The `deploy` directory is where Lucee looks for LEX extension files - when it finds them it will enable that module and remove the file from this directory.

See Lucee documentation for more details on extension deployment.

modules/

Jetty uses modules to group dependencies, libraries, and configuration so that a single module can be enabled without needing to manually enable all the others that it uses, and to verify that relevant files exist.

You may add your own modules to this directory, they will take effect if enabled in `start.ini` file. Jetty's built-in modules reside in `jetty-home/modules/`

You can read more about modules in the Jetty documentation: <http://www.eclipse.org/jetty/documentation/current/startup-modules.html>

modules/lucee.mod

The module definition file for the Lucee module - it lists the other Jetty modules which Lucee depends on to run (`http` for serving requests, `jsp` for servlet functionality).

It also references the main `lucee.jar` to verify it exists at the expected location.

Finally, it references the configuration files that instruct Jetty to handle CFML requests appropriately.

modules/lucee/

The `modules/lucee` directory contains supporting files for the Lucee Jetty module, and the jar file for Lucee.

modules/lucee/lib

This directory contains the Lucee jar file. You may place other JARs in this directory to have them loaded by Jetty and available to Lucee.

modules/lucee/etc/rewrite-rules.xml

This file uses Jetty's [Rewrite Handler](#) in order to allow `.cfm/path_info` URLs to work.

Jetty rewriting supports regex rewrites and redirects, as well as wildcards, plus the ability to set headers and cookies - see documentation at the above Jetty link for further information.

Since v0.7, Lucee on Jetty uses the Jetty rewrite jar but does not enable the Jetty rewrite module itself - you can do that manually if you wish to use it.

modules/lucee/etc/lucee-global-webapp-common.xml

This file uses Jetty's [GlobalWebappConfigBinding](#) to apply [lucee-webapp-common.xml](#) to all context config files.

It is enabled via an entry in [the module definition file](#).

modules/lucee/etc/lucee-webapp-common.xml

This file contains Jetty configuration common to all contexts. Currently this involves only the `overrideDescriptor` required to apply [lucee-servlets.xml](#) and enable the Lucee servlets.

It is applied by the [lucee-global-webapp-common.xml](#) file described above.

modules/lucee/etc/lucee-servlets.xml

This file is a partial web.xml file which configures the Lucee servlets for a webapp context. On its own it doesn't do anything, but when included into a context's XML config file, via `overrideDescriptor`, it activates the Lucee servlets for that context.

(Previous versions of Lucee of Jetty required manually including the `overrideDescriptor` XML directly in the context's config file. This is no longer necessary, it is included automatically via the [lucee-global-webapp-common.xml](#) config.)

There are two Lucee servlets, the main `CFMLServlet` which handles CFML requests, and a `RETServlet` for handling Lucee's RESTful web service functionality.

This file is also where you can define default "welcome" files that are used if a directory is requested - by default it contains `index.cfm` but it also inherits from the defaults in `jetty-home/etc/webdefault.xml` which includes `index.html`, `index.htm` and `index.jsp`

resources/

The resources directory is where Jetty's console-capture module looks for logging configuration, in the form of either `jetty-logging.properties` or `logging.properties` files.

You can read more about logging in the Jetty documentation: <http://eclipse.org/jetty/documentation/current/configuring-logging.html>

resources/jetty-logging.properties

This file configures the logging for Jetty - it has a single line to enable the `StdErrLog` class, which results in `stderr` and `stdout` being redirected to a log file in the `logs` directory.

For more advanced configuration, the Jetty documentation has a [Jetty Logging](#) chapter which also explains how to setup request logging and how to interface with different logging frameworks.

webapps/

This webapps directory is where Jetty WebApp Contexts are located.

There is one context by default - `ROOT.xml` - which listens on localhost and has its `resourceBase` (webroot) set to the `lucee-base/webapps/ROOT` directory.

Jetty's documentation for this is available online: <http://www.eclipse.org/jetty/documentation/current/configuring-deployment.html>

webapps/ROOT.xml

This file configures a webapp context named "ROOT", sets the webroot (`resourceBase`) to `webapps/ROOT/` and specifies that it responds to localhost or 127.0.0.1

The `resourceBase` and `virtualHosts` can be re-configured as desired, or additional contexts can be created by saving new XML files.

New contexts are picked up automatically by Jetty - by default the `deploy` module will check the webapps directory every minute - this can be disabled with `jetty.deploy.scanInterval=0` in `start.ini`, so that contexts are only looked for when the server is started. See the [Hot Deployment](#) documentation for more.

Multiple contexts can run from the same host by using a `contextPath`. Further details at [Setting a Context Path](#).

Virtual Hosts can be fully qualified, use wildcards, IP addresses, connectors, and non-ascii domains via PunyCode. You can also tell Jetty to respond to any hostname that resolves to the machine by removing the `virtualHosts` section. Further information is at [Configuring Virtual Hosts](#).

webinfs/

The webinfs directory contains the configuration for web contexts which Lucee has been enabled for - by default this would be found in a `WEB-INF/lucee` directory in each context's webroot (`resourceBase`), but this bundle has been configured to place them here instead. (Configuration is done via the `CFMLServlet` `lucee-web-directory` parameter in the `etc/lucee-servlets.xml` file.)

Each Jetty context that Lucee is enabled for will have its own directory, named as the `web-context-label` - this defaults to a unique hash but can be changed in Lucee Server Admin.

Unfortunately, Lucee does not allow `web-context-label` to be defined directly in the webapp context configuration, and the label is stored against the hash value. Since the value of the hash is derived simply by hashing the `ServletContext`, any time the context changes (e.g. due to an upgrade), the resulting hash will change, causing a new webinfs directory to be created (whether a label is defined or not), and resetting any per-context configuration.

Common Tasks

Change the port the server runs on

The HTTP server will run on the port specified by the `jetty.http.port` setting in the `lucee-base/start.ini` config file - changes to this file require a Jetty restarted to be picked up.

On Unix-based systems, for historical reasons, ports below 1024 require root permissions, so to listen on the default HTTP port 80, the Jetty documentation has various options at: [Setting Port 80 Access for a Non-Root User](#).

Enable HTTPS requests

To enable HTTPS with Jetty, you need to:

- create a `lucee-base/etc/keystore` file containing SSL certificate data.
- create `lucee-base/etc/jetty-ssl-context.xml` containing `KeyStorePath` and `KeyStorePassword` configuration.
- optionally, override the default port with `jetty.ssl.port=443` in `start.ini` file.
- enable the functionality via the `https` module.
- optionally, enable the `ssl-reload` module to avoid the need to restart when `etc/keystore` is updated.

A walkthrough for doing this with Let's Encrypt certificates is available at: <https://www.sorcerers-tower.dev/articles/configuring-jetty-for-https-with-letsencrypt>

Create a new site on a different domain

For each individual site you should create a new webapp context - there are a few ways to do this with Jetty, but the easiest is to open the `lucee-base/webapps/ROOT.xml` file, modify the `resourceBase` (webroot) and `virtualHosts` (domain aliases) as appropriate, and save it as a new XML file.

Point a new domain at an existing site

This is done by adding items to the `virtualHost` array in a webapp context. For example, the default `ROOT.xml` has this:

```
<Set name="virtualHosts">
  <Array type="String">
    <Item>localhost</Item>
    <Item>127.0.0.1</Item>
  </Array>
</Set>
```

But you might have a context set to:

```
<Set name="virtualHosts">
  <Array type="String">
    <Item>myluceesite.dev</Item>
    <Item>www.myluceesite.dev</Item>
  </Array>
</Set>
```

You can set wildcard subdomains, use punycode for non-ASCII domain names, and more: [Configuring Virtual Hosts](#).

Upgrade Jetty

Upgrading Jetty is as simple as stopping the server, remove or rename the existing `jetty-home` directory, and put the new downloaded Jetty distribution in its place, then start the server again.

(Alternatively, instead of renaming the directories you can simply update the `start-server` script to point at the `start.jar` in the new Jetty directory.)

Upgrade Lucee

Lucee has its own upgrade functionality built-in - either via the Server Admin, or by placing an lco patch in the `lucee-base/lucee-server/patches` directory.

If it is necessary to do a full manual upgrade, you need to stop the server and replace the contents of `modules/lucee/lib` with a new Lucee JAR download.

How do I do something not listed here?

Lucee on Jetty is an on-going project and this documentation is not yet complete.

To prioritise what gets added, please do request missing documentation via the [issue tracker](#), or [via email](#).

Glossary

This glossary aims to define terms you may encounter in the Jetty configuration or documentation which are either not common terms or may have non-typical meanings. If a term is not here (or not fully described), take a look in the Lucee or Jetty docs as appropriate, or consult a dictionary or encyclopaedia. If none of these provide you with a satisfactory answer, please go ahead and raise an issue to get it added.

alias

An alias is an alternate name for something that already has a name. In Tomcat, you can specify aliases for hostname to have multiple domains/subdomains resolve to the same host. In Jetty this is achieved with an array of [virtualHosts](#).

context, webapp, WebAppContext

Within a single server, there can be multiple webapp contexts, where a "webapp" can be seen as a web-based application and a "context" is the area within which that webapp runs. A webapp is not the same thing as a Lucee/CFML application, and multiple CFML applications can run from within a single webapp context.

The `lucee-base/webapps` directory is where Jetty's WebAppContexts are configured. The Lucee configuration for each context is in `lucee-base/webinfs` - the default location is a WEB-INF directory within the webroot, but this is not recommended.

context path, contextPath

The context path can be used to differentiate multiple contexts on a single host - it is a prefix to the [request url](#) that indicates which webapp a URL is referring to.

If all contexts are on their own hosts, the context path is unnecessary and set to an empty string.

Within Lucee, `cgi.context_path` will provide you this value.

distribution

This is a general software term for a piece of completed software. Similar terms are "package" or "bundle". A Jetty distribution is specifically what you get when going to the Jetty website and downloading the software - the filename is in the format `jetty-distribution-{version}.v{timestamp}.{type}`

With Lucee on Jetty, the Jetty distribution is the [jetty-home](#) directory.

host, hostname

In networking terms, a "host" refers to a server, and a hostname is a name for that server. A server can be referred to by multiple names, and thus have multiple hostnames. A domain name is a hostname.

When a domain name refers to an application on a server, but not the server itself, it is considered a *virtual* hostname, or simply virtualhost.

in Jetty, hostnames are configured via the virtualHost property of a [context](#).

JAR

JAR is an abbreviation of Java Archive and a special type of zip file which collects Java classes and other files in a certain structure that can be used by a Java application. A JAR file might contains a small library or an entire piece of software.

`${jetty.home}`, `${jetty.base}`

These are the Jetty variables which refer to the directories containing the home and base aspects of a Jetty setup.

Within Lucee on Jetty, `${jetty.home}` is the `jetty-home` directory, the Jetty distribution, whilst `${jetty.base}` is the `lucee-base` directory.

jetty-home

This is the directory which contains the Jetty distribution. No files within the jetty-home directory are modified for Lucee on Jetty, meaning that upgrading Jetty can generally be done by simply swapping this directory out for an updated one.

lco

This extension refers to a Lucee **CO**re file. These are the files which live in the `lucee-base/lucee-server/patches` directory and are used to upgrade Lucee to a new version.

lex

This extension refers to a Lucee **EX**tension file. These are files which allow you to extend Lucee with additional functionality, and can be installed by placing the file in the `lucee-base/lucee-server/deploy` directory.

lucee-base

This is the directory which configures Lucee to work with Jetty, containing the configuration and overrides to the defaults set by jetty-home plus the Lucee JAR files.

lucee-server

This directory is populated when Lucee first starts, and contains two sub-directories, "context" and "patches". The latter contains [lco](#) files which control which version of Lucee you are running.

The `lucee-server/context` directory is where the server-wide configuration of Lucee is stored. Note that it is *not* an individual context in the Jetty sense.

modules

In Jetty, each piece of functionality is grouped into a module, defined by a `{name}.mod` file in the `jetty-home/modules` directory, which lists the JARs and configuration for the module, plus any other modules the current one depends on to work.

override descriptor, overrideDescriptor

In a Jetty webapp context, the `overrideDescriptor` property refers to a file which contains a partial `web.xml` configuration, which can be used to override the default configuration (`webdefault.xml`) without needing to re-define everything.

Lucee on Jetty uses this to configure the Lucee servlets without repeating the Jetty default servlet configuration.

Previous versions required the `overrideDescriptor` configuration to be placed in each context's XML config, but the current version automatically includes it.

path info, pathInfo

Path Info is the part of a [Request URL](#) that comes after the Script Name. For example, given the request URL `/index.cfm/something/here` the `"/index.cfm"` part is the Script Name whilst the `"/something/here"` part is the Path Info.

It can be accessed in Lucee via `cgi.path_info`.

Note that some servers incorrectly place the *entire* Request URL into the `path_info` variable - this is a bug in that software.

port

In networking, a port is a number on which a server listens for connections. Different protocols default to using different ports - for example, email (SMTP) defaults to port 25 whilst HTTP defaults to port 80.

When a non-default port is used, it is included in the URL after the domain name, separated by a colon, e.g.

`http://domain.com:8080/`

Jetty defaults to running HTTP on port 8080, but this can be configured in start.ini - only one piece of software can listen on each port, so you should ensure that any port you use is not already in use.

Request URL

The Request URL is the part of the URL after the hostname and port, before any query string. It may comprise of Context Path, Script Name, and Path Info parts, or it may contain an arbitrary string which is later re-written into those parts.

requested path, requestedPath

The requestedPath property is how Jetty refers to the Request URL. Jetty only cares about the context path (if any), and then uses its URL mappings to determine which servlet to send the request to. If a URL is rewritten, the requestedPath property is what enables Lucee to determine the original Path Info and ensure `cgi.path_info` is correctly set.

resource base, resourceBase

In Java, a "resource" can refer to a file or directory, and the resourceBase is Jetty's name for the webroot directory (elsewhere known as `public_html` or `htdocs`). This is where the publically-accessible files for a web server should be placed, so that they can be served over HTTP.

resources

Jetty has a "[resources](#)" directory which contains its logging configuration. In this meaning, "resources" is simply saying "files Jetty uses" and is unrelated to the "resourceBase" webroot concept described above.

rewrite

Rewriting a URL is taking one value and changing it to another. This can be done to provide user-friendly URLs rather than the default script-based URLs that webapps traditionally used.

Rewriting can be done with a variety of rules, which use wildcards or regex to identify a URL to be re-written and then replaced with something else. See Jetty's [Rewrite Handler](#) documentation for more information.

Since v0.7, Lucee on Jetty depends on Jetty's rewrite functionality directly, rather than the rewrite module, so to write your own rewrite rules, enable the rewrite module first.

root

Root is a term with multiple related but different meanings:

- root can refer to the highest-level administrator on a machine;
- root can refer to the top-level path on a filesystem (denoted by `/`);
- a webroot (or web-root) is the top-level public directory for a website;
- In Jetty, `ROOT` is the default configuration (it's `contextPath` is a root path);
- In Lucee, `lucee-server-root` is the top-level path for the server configuration;

Server Admin

The Lucee Server Admin allows configuration of values which apply either to all contexts, or that serve as the default values for any new contexts created.

It is available from the URL `/lucee/admin/server.cfm` on all contexts and its files are stored in `lucee-base/lucee-server/context`

server context, server-context

This is a Lucee term that refers to the context directory within the [lucee-server](#) directory. It is only a name - there is not an actual server webapp context.

servlet

A servlet is a server-based application which runs on the JVM. A servlet is executed via a servlet container, and there is a Servlet specification and API which ensures that servlets work the same across different servlet containers.

A Servlet may be a single piece of software, or it may be used to execute other software.

Lucee currently contains two servlets, the main CFML servlet, and a REST servlet (through which RESTful web service functionality is provided).

servlet container, servlet engine

A servlet container is any software which runs a [servlet](#) in accordance with the [Servlet specification](#).

Jetty and Tomcat are the primary examples of servlet containers, but both of them are more than just containers.

stderr, stdout

Standard Error (stderr) and Standard Output (stdout) are the two destinations a command line application can send its output to. stderr is for any messages which are considered errors or failures, whilst stdout is any other output.

In different situations these may be redirected to different location - when interacting with software, you want errors on screen but might want the output to a file.

For server software, such as Jetty, it generally makes sense to have everything go to a log file instead of on screen (since many servers don't have screens), which means redirecting stderr and stdout so they are appended to a log file.

virtual host, virtualHosts

A virtual host is the term for when a [hostname](#) does not refer to a real server, but instead refers to an application on a server with a different name. This can sometimes be called an "alias", or a "parked domain" by hosting service providers.

In Jetty, virtual hosts are configured per-context using the virtualHost property, which can contain an array of domains, IP addresses, etc to which that context will respond.

Web Admin

Lucee provides a Web Admin for each webapp context, allowing you to configure them each individually. (There is also a single [Server Admin](#) which can be used to set global defaults.)

The Web Admin is available at `/lucee/admin/web.cfm` and saves its configuration within the `lucee-base/webinfs/{web-context-label}/` directory (where a context's label defaults to a unique hash and can be configured in the Server Admin).

webapps, WebApplicationContext

See [context](#).

web-context-label, web-context-hash

Lucee allocates a unique `web-context-hash` to each context, plus a customisable `web-context-label` (which defaults to the `web-context-hash`). These variables can be used within servlet configuration parameters to define where the Lucee configuration directory goes.

WEB-INF, webinfs

A WEB-INF directory is the default location for a context's configuration, and will most often be seen if you look inside a .WAR file.

Lucee, by default, creates a `WEB-INF/lucee` directory where Lucee's configuration is stored, but for Lucee on Jetty, the location of this directory has been overridden to `lucee-base/webinfs/{web-context-label}` which takes it out of the webroot (and means that a WEB-INF directory does not get created, unless some other servlet creates it).

webroot

The webroot (also known as htdocs, public_html, wwwroot) is the directory on a server where public-facing files are placed.

In Jetty it is called the [resourceBase](#) and can be configured independently for each servlet (i.e. you do not need to place the webroot for multiple contexts within the same parent directory, though you can if you wish).

Lucee on Jetty default configuration has the webroot at `lucee-base/webapps/ROOT`